Attorney Docket No.: 20155-000100US

PATENT APPLICATION

SYSTEM USING INDIRECT MEMORY ADDRESSING TO PERFORM CROSS-CONNECTING OF DATA TRANSFERS

Inventor: Patrick Christian
a citizen of the UK residing at
2 Clare Mead, Rowledge, Farnham
Surrey, GU10 4BJ
UNITED KINGDOM

Assignee: VegaStream Ltd.
Technology Transfer Center
Silwood Park, Buckhurst Road
Ascot, Berkshire SL5 7PW
ENGLAND

Entity: Small

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California 94111-3834
Tel: 415-576-0200

SYSTEM USING INDIRECT MEMORY ADDRESSING TO PERFORM CROSS-CONNECTING OF DATA TRANSFERS

~~CROSS-REFERENCE TO RELATED APPLICATION~~

This application claims priority from British Patent Application No. 0002062.8 filed on January 28, 2000, the disclosure of which is incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

This invention relates in general to digital systems and more specifically to a digital system using indirect memory addressing to perform cross-connecting of data streams.

Digital data formats are being used to represent almost every type of information imaginable. Not only are numbers, text and images represented digitally, but digital formats now include standards for voice and video. For example, standards promulgated by the International Telecommunications Union (ITU) provide standard specifications for compressing and transferring digital audio and video. The use of digital formats has distinct benefits in providing information that can be easily and accurately stored, transferred, processed and presented to human end users or electronic digital systems.

In some applications, such as telephony or video, it is desirable to provide a continuous "stream" of data in real time. This is necessary in cases where the digital data must be transferred immediately (as with telephony) or where the amount of data is so large that it is more efficient to present the data immediately as it is being received so that large buffers or other extensive storage is not necessary (as with video). Such streaming data not only needs to be transferred as quickly and as efficiently as possible, but it must be sent to particular destinations over a network. When the number of users of a network is very large, the problem of handling fast streams of data efficiently while also providing the ability to quickly and accurately deliver the streaming data to desired end users becomes complex.

The world-wide Internet has become a popular network. A great deal of effort is being focused on inventions to allow the Internet to handle streaming data while

still maintaining the Internet benefits of a flexible routing scheme and the ability to massively scale to millions of users and content providers. These abilities should allow the Internet to be successfully adapted to such applications as telephony, video, three-dimensional simulation, email, or other audio and image digital data distribution applications.

However, the nature of the Internet's "Internet Protocol" (IP) and distributed routing requires that data, and data streams, be divided into many small "packets" of information. These packets of information must be directed, or switched, at near wire-speed without undue delay. Such a switching system must be extremely flexible in handling point-to-point, point-to-multipoint, or other possible permutations of data switching. Because of the distributed nature of the Internet, many such switches are required at many points so it is necessary to make the switches operate with as little resources (e.g., memory, processing power) as possible while still achieving the desired performance.

Thus, it is desirable to provide a fast, flexible switching system that efficiently handles data transfers in a network.

SUMMARY OF THE INVENTION

The present invention uses indirect memory addressing to perform switching of digital data streams in a network. Incoming data is organized into timeslots. Each timeslot's data is stored into a predefined location in buffer memory. Indirect addressing is implemented in a crosspoint address table. The addresses stored in the crosspoint address table are used to access locations in the buffer memory so that portions of words stored in the buffer memory can be combined to form an output word destined for a predetermined timeslot.

In a preferred embodiment, 32-bit words are stored in the buffer memory. The indirect addressing allows any byte of any word in the buffer memory to be accessed and used to form an outgoing 32-bit word. The system operates on clock cycles whereby a word of incoming data is stored at the beginning of each clock cycle and four addresses are fetched from the crosspoint address table and used to access four bytes of buffer data by the end of each clock cycle. Thus, there is a word of data coming in and a word of data going out on each clock cycle. Both the buffer storage and crosspoint address table are accessed sequentially.

One embodiment of the invention provides a system for transferring data from an incoming source to an outgoing destination. The system includes buffer storage coupled to the incoming source for receiving and storing data from the incoming source; a buffer storage address generator for sequentially addressing the buffer storage so that incoming data is stored sequentially within the buffer storage; an address storage including one or more addresses for accessing the buffer storage; an address storage address generator for sequentially accessing the one or more addresses stored in the address storage; and an output stage, wherein the output stage retrieves data from the buffer storage in accordance with the one or more addresses accessed by the address generator.

Another embodiment of the invention provides a method for switching digital data streams in a network where the digital data streams include timeslots. The method includes storing incoming data in a memory in accordance with the incoming data's timeslot and indirectly accessing the memory to determine which portions of the data to output.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a diagram illustrating the system of the present invention.

DESCRIPTION OF THE SPECIFIC EMBODIMENTS

Fig. 1 is a diagram illustrating the system of the present invention.

In Fig. 1, incoming data streams 110 can include data from one or more data streams. The data streams can be serial or parallel streams. The streams can be of varying data width. Serial to parallel converter 112 is used to form data word 114 for storage. Typically, the data output from the data streams is designed so that one 32-bit word is available in a "timeslot." A timeslot is an arbitrary interval of time for use in time-division-multiplexed (TDM) channels such as Mitel's ST-Bus. A popular mode would be to take 8 bits of data (one "byte" of data) from each of four streams in a single timeslot and to combine the bytes into a 32-bit word. Note that other input schemes are possible. The present invention is adaptable, in general, to any application where TDM or TDM-like communication channels are used.

Data word 114 is stored into buffer memory 118. In a preferred embodiment, buffer memory 118 is a subsection of total memory 116. However, any memory architecture may be used. For example, memory 118 may be an entire bank of memory on a separate integrated circuit chip, may be "virtual" memory on a hard disk drive or other media, may be arranged as any word, length; may be dual-port memory, etc.

Counter 128 generates addresses for the storage of data word 114 into buffer memory 118. In Fig. 1, data word 114 is shown being stored into location 138. Other examples of data words stored into memory 118 are shown at 130, 132, 134 and 136. Counter 128 increments sequentially, 4 bytes at a time, and is synchronized to the timeslots of the incoming data. Since the timeslot data arrives in sequence, each location in buffer memory corresponds to a single timeslot, in sequence. Thus, for example, buffer location 0 corresponds to timeslot 0, buffer location 1 corresponds to timeslot 1, and so on. Note that, where buffer storage 118 is part of a larger memory bank, a "base address" serves to define location 0 of the buffer storage. As is known in the art, an index into the buffer is used as the basis for accessing word locations in the buffer storage. With this design, each incoming data word is stored sequentially into the buffer storage and buffer storage locations correspond to timeslots.

Note that other incoming storage mappings are possible. For example, the incoming data words can be stored in descending order. The storage can be in reverse correspondence to the timeslot numbers. A hashing function can be implemented to map timeslots to arbitrary locations, words can be stored in multiple storage buffer locations, etc. In general, any suitable incoming storage mapping may be employed

Fig. 1 shows crosspoint address table 160 also occupying total memory 116. As described above regarding the storage buffer, other memory architectures are possible. Crosspoint address table 160 includes pre-stored addresses for accessing bytes residing in buffer storage 118. The pre-stored addresses are typically written prior to a switching session by a host processor (not shown). The pre-stored addresses define the routing of incoming timeslot data to outgoing timeslot data. Since the addresses are byte-specific they can point to any byte in any word of buffer storage 118. Thus, an outgoing word can include any byte from any of the incoming timeslots.

On each cycle, four addresses are read from crosspoint address table 160. For example, in a given cycle, addresses 162, 164, 166 and 168 are read. These addresses reference bytes 134, 132, 136 and 130, respectively. The referenced bytes are read from

storage buffer 118 and combined in byte extractor 120 to form 32-bit word 122. When it is desirable to split the outgoing data into multiple streams then parallel to serial converter 124 can be used to generate, e.g., 4 outgoing data streams 126, as shown.

Crosspoint address table 160 is accessed sequentially. In the preferred embodiment, four addresses are accessed each clock cycle, or timeslot. Naturally, any number of accesses can be employed. Also, the same options as to memory size and implementation as discussed above with respect to the buffer storage are possible.

It should be apparent that the system of the present invention can be used to arbitrarily assign any incoming data to an outgoing timeslot. Also, incoming data can be assigned to multiple outgoing timeslots. Data can be "dropped" or ignored, etc. Although the host processor in the preferred embodiment only writes to the crosspoint address table prior to the actual switching session, dynamic updating of the crosspoint address table is possible. Although the system of the present invention has been described with respect to a specific hardware configuration, many variations are possible. Components can be combined, omitted, or added. Functions can be implemented in hardware, software, or a combination of hardware and software. Incoming and outgoing data transfers need not be synchronized with each other, as where the number of outgoing timeslots differs in number from the incoming timeslots. Other variations are possible.

Although the present invention has been described with reference to specific embodiments thereof, these embodiments are merely illustrative, and not restrictive, of the invention, the scope of which is determined solely by the appended claims.